package com.twitter.home\_mixer.marshaller.request

import com.twitter.home\_mixer.model.request.HomeMixerRequest

import com.twitter.home\_mixer.{thriftscala => t}

import com.twitter.product\_mixer.core.functional\_component.marshaller.request.ClientContextUnmarshaller

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class HomeMixerRequestUnmarshaller @Inject() (

clientContextUnmarshaller: ClientContextUnmarshaller,

homeProductUnmarshaller: HomeMixerProductUnmarshaller,

homeProductContextUnmarshaller: HomeMixerProductContextUnmarshaller,

homeDebugParamsUnmarshaller: HomeMixerDebugParamsUnmarshaller) {

def apply(homeRequest: t.HomeMixerRequest): HomeMixerRequest = {

HomeMixerRequest(

clientContext = clientContextUnmarshaller(homeRequest.clientContext),

product = homeProductUnmarshaller(homeRequest.product),

productContext = homeRequest.productContext.map(homeProductContextUnmarshaller(\_)),

// Avoid de-serializing cursors in the request unmarshaller. The unmarshaller should never

// fail, which is often a possibility when trying to de-serialize a cursor. Cursors can also

// be product-specific and more appropriately handled in individual product pipelines.

serializedRequestCursor = homeRequest.cursor,

maxResults = homeRequest.maxResults,

debugParams = homeRequest.debugParams.map(homeDebugParamsUnmarshaller(\_)),

homeRequestParam = false

)

}

}